An Efficient Training Approach for Very Large Scale Face Recognition

Kai Wang[1, 2*] Shuo Wang [2*] Panpan Zhang[1] Zhipeng Zhou[2] Zheng Zhu[3]
Xiaobo Wang[4] Xiaojiang Peng[5] Baigui Sun[2] Hao Li [2] Yang You[1†]
[1]National University of Singapore [2]Alibaba Group [3]Tsinghua University
[4]Institute of Automation, Chinese Academy of Sciences [5]Shenzhen Technology University

Code: https://github.com/tiandunx/FFC

Abstract

Face recognition has achieved significant progress in deep learning era due to the ultra-large-scale and well-labeled datasets. However, training on the outsize datasets is time-consuming and takes up a lot of hardware resource. Therefore, designing an efficient training approach is indispensable. The heavy computational and memory costs mainly result from the million-level dimensionality of the fully connected (FC) layer. To this end, we propose a novel training approach, termed Faster Face Classification (F^2C), to alleviate time and cost without sacrificing the performance. This method adopts Dynamic Class Pool (DCP) for storing and updating the identities' features dynamically, which could be regarded as a substitute for the FC layer. DCP is efficiently time-saving and cost-saving, as its smaller size with the independence from the whole face identities together. We further validate the proposed F^2C method across several face benchmarks and private datasets, and display comparable results, meanwhile the speed is faster than state-of-the-art FC-based methods in terms of recognition accuracy and hardware costs. Moreover, our method is further improved by a well-designed dual data loader including indentity-based and instance-based loaders, which makes it more efficient for updating DCP parameters.

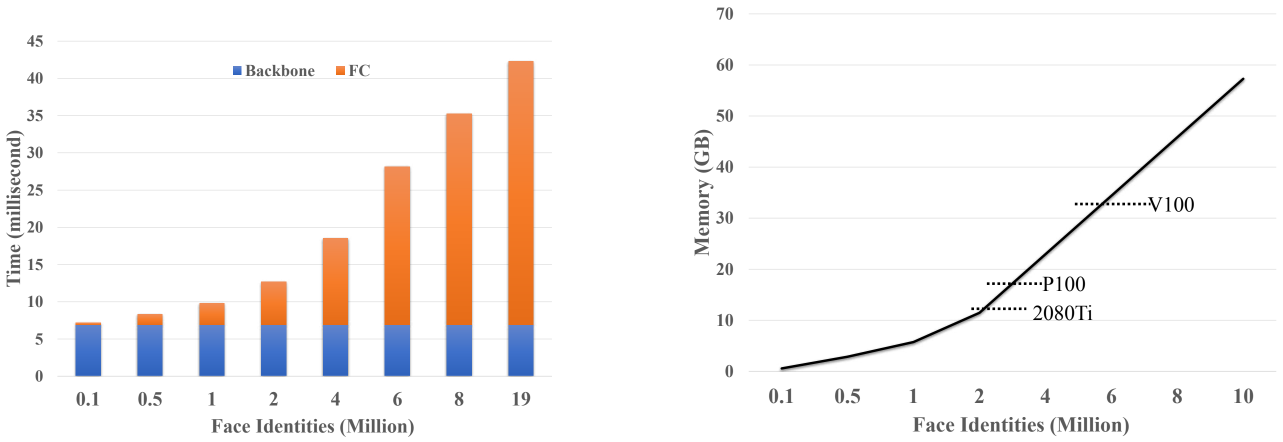
1. Introduction

Deep Neural Networks (DNNs) has achieved many remarkable results in computer vision tasks [4, 6, 7, 26, 27, 37, 38, 39, 40]. Face recognition can be regarded as one of the most popular research topics in computer vision. Many large scale and well-labelled datasets have been released over the past decade [11, 15, 47, 49, 53]. The training process of face recognition aims to learn identity-related embedding space, where the intra-class distances are reduced and inter-class distances are enlarged in the meanwhile. Previous works [11, 42, 43] have proved that training on a large dataset can obtain a substantial improvement over a small dataset. To this end, academia and industry collected ultra-large-scale datasets including 10 even 100 million face identities. Google collected 200 million face images consisting of 8 million identities [28]. Tsinghua introduced WebFace260M [53] including 260 million faces, which is the largest public face dataset and achieves state-of-the-art performance.

In general, these ultra-large-scale datasets boost the face recognition performance by a large margin. However, with the growth of face identities and limitations of hardware, there are mainly two problems in training phase. The first problem results from the training time and hardware resource occupancy. As shown in Fig 1, the time cost and GPU memory occupancy of the FC layer are much greater than those of the backbone when the face identities reach 10 million. To address these issues, many previous methods [1, 20] focus on reducing the time and resource cost of the FC layer. Previous methods can be summarized into two categories. One [1] tries to distribute the whole FC to different GPUs, introducing heavy communication costs. The other [50] attempts to reduce the computing cost by selecting a certain ratio of neurons from the FC layer randomly, but it still needs to store the whole FC parameters. When the identities reach 10 or 100 million, storing the whole FC parameters is extremely expensive. How to effectively reduce the computational and memory costs caused by the high-dimensional FC layer? An intuitive idea is to decrease the size of FC or design an alternative paradigm, which is hardly explored before. The second problem is related to the update efficiency and speed of FC parameters. As pointed by [9], the optimal solution for the class center is actually the mean of all samples of this class. Identities that have rare samples with very low frequency of sampling will have very little opportunity to updat class centers through their

[*] Equal contribution. (kai.wang@comp.nus.edu.sg, wangshuo514@sina.com)

[†] Corresponding author (youy@comp.nus.edu.sg).



(a) Comparison of backbone and FC time cost (ms).

(b) The memory occupancy of the FC layer at training phase (G).

Figure 1: Visualization of training time and GPU memory occupancy. Figure 1a shows the forward time comparison of backbone (ResNet50) and the FC layer. Given an image, the time cost of FC increases sharply with the growing number of face identities but the time of backbone stays unchanged. Figure 1b illustrates the GPU memory occupancy with the size of face identities. Even the V100 32G GPU can only store the FC parameters with the output size of about 6 millions (The dimension of face recognition is usually 512). Therefore, it is very necessary to design a method that reduces the training time and hardware cost of the FC layer.

samples, which may hamper feature representation.

To tackle aforementioned issues, we propose an efficient training approach for ultra-large-scale face datasets, termed as Faster Face Classification (F^{2}C). In F^{2}C, we first introduce twin backbones named Gallery Net (G-Net) and Probe Net (P-Net) to generate identity centers and extract face features, respectively. G-Net has the same structure with P-Net and inherits the parameters from P-Net in a moving average manner. Considering that the most time-consuming part of the ultra-large-scale training lies at the FC layer, we propose Dynamic Class Pool (DCP) to store the features from G-Net and calculate the logits with positive samples (whose identities appear in DCP) in each mini-batch. DCP can be regarded as a substitute for the FC layer and its size is much smaller than FC, which is the reason why F^{2}C can largely reduce the time and resource cost compared to the FC layer. For negative samples (whose identities do not appear in the DCP), we minimize the cosine similarities between negative samples and DCP. To improve the update efficiency and speed of DCP parameters, we design a dual data loader including identity-based and instance-based loaders. The dual data loader loads images from given dataset by instances and identities to generate batches for training. Finally, we conduct sufficient experiments on several face benchmarks to prove F^{2}C can achieve comparable results and a higher training speed than normal FC-based method. F^{2}C also obtains superior performance than previous methods in term of recognition accuracy and hardware cost. Our contributions can be summarized as follows.

1) We propose an efficient training approach F^{2}C for ultra-large-scale face recognition training, which aims to reduce the training time and hardware costs while keeping comparable performance to state-of-the-art FC-based methods.

2) We design DCP to store and update the identities' features dynamically, which is an alternative to the FC layer. The size of DCP is much smaller than FC and independent of the whole face identities, so the training time and hardware costs can be decreased substantially.

3) We design a dual data loader including identity-based and instance-based loaders to improve the update efficiency of DCP parameters.

2. Related Work

Face Recognition. Face recognition has witnessed dramatical progress due to the large scale datasets, advanced architectures and loss functions. Large scale datasets play the most crucial role in promoting the performance of face recognition [8]. These datasets can be divided into three intervals according to the number of face identities: 1-10K, 11-100K, >100K. VGGFace [25], VGGFace2 [3], UMDFaces [2], CelebFaces [32], and CASIA-WebFace [49] belong to the first interval. The face identities of the IMDBFace [34] and MS1MV2 [8] are between 11K to 100K. Glint360k [1] and Webface260M [53] have about 0.36M and 4M identities. Many previous works [1, 14, 50, 53] illustrate that training on larger face identities datasets can achieve better performance than on smaller ones. Therefore using WebFace260M as the training dataset obtains state-of-the-art performance on IJBC[23] and top 3 in NIST-FRVT challenge. Based on these datasets, a variety of CNN architectures for improving the performances, such as VGGNet [30], GoogleNet [33], ResNet [13], AttentionNet [36] and MobileFaceNet [5], have been proposed. For the loss function, contrastive loss [32, 48] and triplet loss [30] might be good candidates. But they suffer from high computational cost and slow convergence. To this end, researchers attempt to explore new metric learning loss functions to boost the face recognition performance. Several margin-based softmax losses [8, 21, 35, 44, 45] have been exploited and obtained the state-of-the-art results. To sum up, current methods and large scale datasets have achieved excellent performance in face recognition, but the training time and hardware costs are still the bottleneck at training phase, especially for training on million scale or even more face identities datasets.

Acceleration for Large-Scale FC Layer. As illustrated in Figure 1a, the time cost mainly focuses on FC layer rather convolutional layer when the face identities reach 10M. Researchers try some attempts to accelerate the large scale FC training since 2001. An intuitive idea is to design an approximate function to reduce the computational cost, the Hierarchical Softmax (HSM)[10] tries to reformulate the multi-class classifier into a hierarchy of binary classifiers. Therefore, the training cost can be reduced by means of the given sample only has to traverse along a path from the root to the corresponding class. However, all the class centers are stored in RAM and the retrieval time can not be ignored with the increase of face identities. Zhang *et.al.* [50] proposed a method that can recognize a small number of **"active classes"** in each mini batch, which constructs the dynamic class hierarchies on the fly. However, recognizing the **"active classes"** is also time-consuming when the face identities is too large. Some companies, such as Google and Microsoft, try to divide all the categories into multi-GPUs averagely. The communication cost of inter-servers can not be ignored. To tackle this problem, Partial FC [1] tries to train a large-scale dataset on a single GPU server using 10% identities randomly at each iteration. However it's still limited by the memory of the GPUs in a single machine. As shown in Figure 1b, Partial FC can only work when the number of face identities is not ultra-large (<10M), otherwise the GPUs will still run out of memory. There are several pairwise based methods [16] that utilize the face pairs to train large scale datasets, while the time complexity is $O(N^k)$, where k represents the size of the pair. The latest related work VFC [20] builds some virtual FC parameters to reduce the computation cost but its performance is much lower compared to normal FC. Different from previous works, our F^{2}C can reduce the FC training cost largely and achieve comparable performance compared to normal FC-based methods.

3. Faster Face Classification

In this section, we first give an overview of F^{2}C for a brief understanding of our method. Then we present our motivation and key modules for ultra-large-scale datasets training. After that, we show the theoretical/empirical analysis over these modules. Finally we demonstrate the training details for better reproduction.

3.1. Overview of F^{2}C

The problem we tackle is to accelerate the training speed and reduce the hardware costs of ultra-large-scale face datasets (face identities > 10M) without obvious degradation of performance. To this end, we propose F^{2}C framework for ultra-large-scale face datasets training. As shown in Figure 2, given ultra-large-scale face datasets, we utilize instance-based loader to generate an instance batch as data loader usually does. Meanwhile, identity-based loader selects two images randomly from the same identity to form the paired identities batch. Subsequently, we mix up the images from instance and pair identity batches as shown in Figure 2 and feed them into G-Net and P-Net. Inspired by MoCo [12], G-Net has the same structure as P-Net and inherits parameters from P-Net in a moving average manner. G-Net and P-Net are used to generate identities' centers and extract face features for face recognition, respectively. Then, we introduce DCP as a substitute for the FC layer. DCP is randomly initialized and updated by the features from G-Net at each iteration. The update strategy of DCP follows the rule: using the current features to replace the most outdated part of features in DCP. For positive samples, we use the common cross entropy loss. For negative samples, we minimize the cosine similarities between negative samples and DCP. The whole F^{2}C is optimized by cross entropy loss and cosine similarities simultaneously.

3.2. Motivation

Before digging into F^{2}C, we provide some motivations by rethinking the loss function cooperated with FC layer. For convenience, we consider the Softmax as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n_{\text{ID}}} e^{W_j^T x_i}} \tag{1}$$

where N is batchsize and n_{ID} stands for the number of whole face identities. For each iteration of the training process, the update of the classifier $\{W_j\}_{j=1}^{n_{\text{ID}}}$ is performed as the following equations:

$$\frac{\partial L}{\partial W_k} = -\frac{1}{N}\sum_{i=1}^{N}(\delta_{ky_i} - \frac{e^{W_k^T x_i}}{\sum_{j=1}^{n_{\text{ID}}} e^{W_j^T x_i}})x_i \tag{2}$$

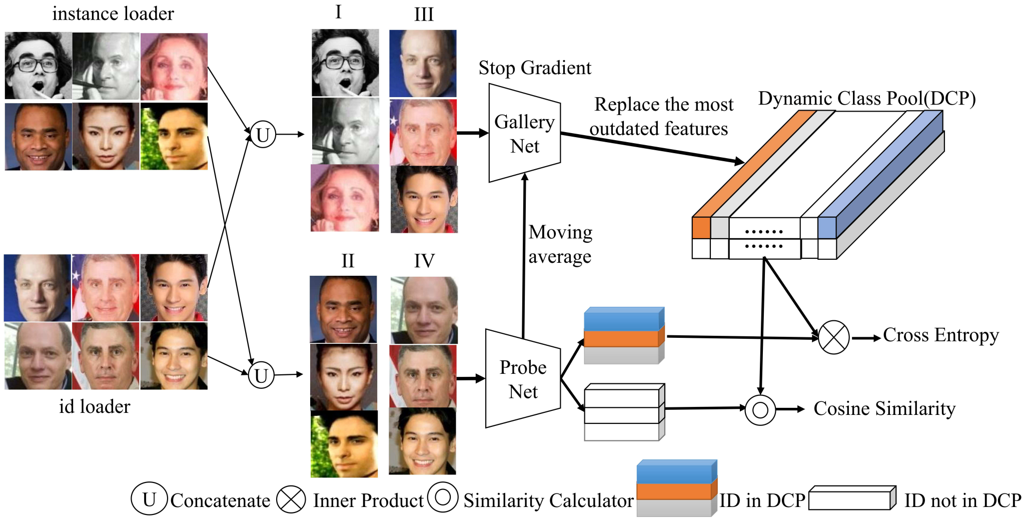


Figure 2: The pipeline of F^{2}C. We use instance and id data loader to generate mixed batches (I ∪ III, II ∪ IV), which are later fed into G-Net and P-Net respectively. The features from G-Net will update DCP in the manner of LRU, and features from P-Net will be used to compute loss together with DCP.

Obviously, all the classifiers $\{W_j\}_{j=1}^{n_{\rm ID}}$ will be updated in each iteration, which means each classifier has the same chance to be optimized. The goal of face recognition is to distinct persons from different identities with the mechanism where the features from the same identity are pulled together and features belonging to different identities are pushed away. As the main problem of training with ultra-large-scale dataset is the explosive size of FC layer, We can consider the whole FC as a set of classifiers. In order to reduce the computation cost, it is intuitive for us to optimize fixed ratio of the classifiers in each iteration during the training process. Specifically, we utilize a vector as follows to represent whether a given classifier is in optimization queue.

$$V = \{\nu_1, ..., \nu_{n_{\rm ID}}\}, \forall i, \nu_i \in \{0, 1\} \text{ and } \#\{\nu_i|\nu_i \neq 0\} = C \tag{3}$$

where C is a constant stands for the length of the optimization queue, $\nu_i = 0/1$ denotes the classifier W_i is (not in)/(in) optimization queue. We draw the corresponding objective for this setting.

$$\hat{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n_{\rm ID}} \nu_j e^{W_j^T x_i}} \tag{4}$$

The classifiers update on basis of the following equations:

$$\frac{\partial \hat{L}}{\partial W_k} = -\frac{1}{N}\sum_{i=1}^{N}(\delta_{ky_i} - \frac{\nu_k e^{W_k^T x_i}}{\sum_{j=1}^{n_{\rm ID}} \nu_j e^{W_j^T x_i}})x_i \tag{5}$$

Formally, equation 5 is similar to equation 2, the selection mechanism for vector V will influence the update process of the classifier directly. We should design feasible selection mechanism for better optimization of classifiers under the constraint that only partial classifiers will be updated at each iteration. However, this straightforward method still suffer the heavy pressure from storing the whole set of classifiers. As a matter of fact, in our novel framework, we only offer limited space to store a fixed ratio of classifier/features dynamically.

3.3. Identity-Based and Instance-Based Loaders

In this subsection, we introduce the details of our dual data loader. For convenience, we denote the batchsize as **M**. Practically, we utilize the instance-based loader to sample **M** images from a given face dataset randomly to get the instance batch. In the meanwhile, the identity-based loader is applied to provide identity batch by selecting M identities randomly without replacement from the whole identities and sampling two images for each identity. We divide the instance batch into two parts, with **M/2** images for each part. For paired identity batch, we split it by face identity to form two parts with same set of face identities. We mix up the four parts to get $I \cup III; II \cup IV$ (as illustrated in Figure 2), where ∪ represents the union operation for sets.

Why Dual data loaders? As aforementioned, we design dual data loader to improve the update efficiency of DCP parameters. To better understand our design, we analyze the different influences between identity-based and instance-based loaders as follows. Let M denote batch size, n_{ID} be the total number of identities of the given the dataset, k_{min} (k_{max}) as the minimum (maximum) number of images for one person in the dataset, $\bar{k}$ be the average number of images per identity. Here the shape of DCP mentioned in the main paper is $C \times K \times D$. C is the magnitude of DCP, K is the capacity for each placeholder in DCP, D represents feature dimension. The total images of given dataset can be denoted as $\bar{k}n_{ID}$. We evaluate the update speed by estimating the minimum of epochs for given face identity to update $\frac{\bar{k}n_{\text{ID}}}{M}$.

- If we only use instance-based loader, the update speed of identities' centers $\in [\frac{\bar{k}n_{ID}}{Mk_{max}}, \frac{\bar{k}n_{ID}}{Mk_{min}}]$. So only using instance-based loader may lead to following problems. 1. If the number of identities is severely imbalanced, the update speed of the identities' centers that have rare number of images is too slow. 2. If we sample M images that belong to M different identities, the DCP may have no positive samples for this iteration. In this case, cross entropy, which is crucial for classification, cannot be calculated.

- If we only use identity-based loader, we can obtain the average fastest update speed ($\frac{n_{ID}}{M}$) of each identity. However, identity-based loader re-sample identities that have rare number of images too many times, so it needs to use about $\frac{k_{max}}{k_{min}}$ times more iterations than instance-based loader to sample all images from the dataset. Further, the sample probabilities for each instance of identities with rich intra-class images are too low, the identity-based loader can not sample plenty of intra-class images during the training phase.

- Using the dual data loader can inherit the benefits from instance-based and identity-based loaders. First, dual data loader provides appropriate ratios between positive and negative images, which is very important for DCP. Second, dual data loader keeps high update efficiency (speed) of identities' centers and various intra-class images.

Feature Extraction We take $I \cup III$ and $II \cup IV$ as input to Probe and Gallery Nets respectively to extract the face features and generate the identities' centers. The process can be formulated as follows:

$$\begin{aligned} P_\theta(I \cup III) &= F_p^{\text{DCP}} \oplus F_p^{\neg\text{DCP}} \\ G_\phi(II \cup IV) &= F_g \end{aligned} \tag{6}$$

where the probe and gallery net are abbreviated as P_θ and G_ϕ with parameters denoted as θ, ϕ respectively. The symbol $\neg$ is set to split features whose identities belong to DCP (subsection 3.4) from those do not. F_g represents the features extracted by the Gallery Net. For each batch, we denote number of identities in DCP as I and number of identities not in DCP as $M - I$.

3.4. Dynamic Class Pool

In this subsection, we introduce the details of the Dynamic Class Pool (DCP). Inspired by sliding window [18] in object detection task, we can utilize a dynamic identity group that slides the whole face identities by iterations.

We called this sliding identity group as DCP, which can be regarded as a substitute for the FC layer. Firstly, we define a tensor T with size of $C \times K \times D$ which is initialized with Gaussian distribution, where C is the capacity or the number of face identities the DCP can hold, K represents the number of features that belong to the same identity (we set the default as $K = 2$). We store F_g in DCP and update the most outdated features of DCP using the F_g in each iteration. The updating rule is similar to least recently used (LRU)[1] policy which can be formulated as,

$$\begin{aligned} &T[1 : C - M, :, :] = T[M + 1 : C, :, :] \in \mathbb{R}^{(C-M)\times K\times D} \\ &T[C - M + 1 : C, 0, :] = F_g \in \mathbb{R}^{M\times K\times D} \end{aligned} \tag{7}$$

For the current batch, with the update of the DCP, we obtain pseudo feature center for each identity in DCP, including the identities contained in $II \cup IV$. As claimed in equation 6, features from P-Net can be divided into two types compared to DCP. One is F_p^{DCP}, the other is $F_p^{\neg\text{DCP}}$. For F_p^{DCP}, we can calculate its logits by the following equation,

$$P = \frac{1}{K}\sum_{i=1}^{K} \langle F_p^{\text{DCP}}, T[:, i, :] \rangle \in \mathbb{R}^{I\times C} \tag{8}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product operation, P represents the logits of F_p^{DCP}. Therefore, we can formulate the Cross-Entropy loss as follows:

$$L_{\text{ce}} = -\frac{1}{I}\sum_{i=1}^{I} \log \frac{e^{W_{y_i}^{\text{T}} P_i}}{\sum_{j=1}^{C} e^{W_j^{\text{T}} P_i}}, \tag{9}$$

where W_j is the j-th classifier, y_i is the identity of P_i. For features $F_p^{\neg\text{DCP}}$ whose IDs are not in DCP, we add a constraint to minimize the cosine similarity between $F_p^{\neg\text{DCP}}$ and T, which can be formulated as,

$$L_{\text{cos}} = \frac{1}{M - I}\sum_{i=1}^{M-I} \varphi(F_p^{\neg\text{DCP}}, \bar{T}), \tag{10}$$

where φ is the operation of calculating the cosine similarity, $\bar{T}$ represents the average operation along the axis of K in DCP. The total loss is $L_{\text{total}} = L_{\text{ce}} + L_{\text{cos}}$.

3.5. Empirical Analysis.

DCP As shown in equation 4 and 9, the cross entropy loss we utilize for DCP is similar to the loss for FC formally. With special setting of vector V in equation 3, we can represent L_{ce} in the form of equation 4. For further verification of the effect of this mechanism on the training with DCP, we provide some empirical analysis.

[1] https://www.interviewcake.com/concept/java/lru-cache

Algorithm 1: Update Mechanism of DCP

```
Input:
DCP: $T \in \mathbb{R}^{C \times K \times D}$ initialized with Gaussian distribution.
Index for the identity batch: $t$.
Batch Size: $M$.
for $1 \leq t \leq \frac{n_{ID}}{M}$ do
    utilize the G-Net to extract features from $t$-th batch as the pseudo feature centers denoted as $F_g$;
    if $1 \leq t \leq \frac{C}{M}$:
        store $F_g$ sequentially in those unoccupied position in DCP.
    else:
    update DCP as shown in Equation 7
end
```

As mentioned in subsection 3.3 and equation 7, the identities in DCP are updated in an LRU mechanism as shown in Algorithm 1. As identity-based loader goes through the dataset in terms of identities, partial components($\frac{M}{2}$) of vector V can be determined by shuffling the whole face identities and taking the corresponding t-th part of it, where $1 \leq t \leq \frac{n_{ID}}{M}$. When we use identity-based loader, then by the setting of V and property of LRU rules, each classifier/pseudo feature center can be updated at least $[\frac{C}{M}]$ times. This means that every classifier can have the similar chance to be optimized in our settings. DCP may have the following benefits: 1) The size of DCP is independent from magnitude of face identities, which can be far smaller than FC. Therefore the computational cost is greatly reduced; 2) The hardware especially storage occupancy of DCP is also smaller than FC and the communication cost can be reduced dramatically. These benefits are the reasons why we call our method as Faster Face Classification.

3.6. Experimental Details

We train our F^{2}C on a single server with 8 Tesla V100 32G GPUs. We utilize ResNet100, ResNet50 and MobileFaceNet as our backbones to evaluate the efficiency of F^{2}C. The learning rate is initialized as 0.1 with SGD optimizer and divided by 10 at 10, 14, 17 epochs. The training is terminated at 20 epochs. The length (number of ID) of DCP is defaulted as 10% of total face identities. The batch size is 512 *i.e.*, 256 images from identity-based loader and 256 images from instance-based loader.

4. Experiments

In this section, we first review several benchmark datasets in face recognition area briefly. Then, we conduct ablation studies to evaluate the effectiveness of each module and the settings of hyper-parameters in F^{2}C. Finally, we compare F^{2}C to related state-of-the-art methods.

4.1. Datasets

We utilize MobileFaceNet, ResNet50 and ResNet100 to train F^{2}C on MS1MV2, Glint360k and Webface42M (Webface42M is the cleaned version of the original Webface260M and it has 2M ID and about 42M images), respectively. We mainly show the performance of F^{2}C in following 9 academic datasets: LFW [15], SLFW [15], CFP [29], CALFW [52], CPLFW [51], AGEDB [24], YTF [46], IJBC [23], and MegaFace [17]. LFW is collected from the Internet which contains 13,233 images with 5,749 IDs. SLFW is similar to the LFW but the scale of SLFW is smaller than LFW. CFP collects celebrities' images including frontal and profile views. CALFW is a cross-age version of LFW. CPLFW is similar to CALFW, but CPLFW contains more pose variant images. AGEDB contains images annotated with accurate to the year, noise-free labels. YTF includes 3425 videos from YouTube with 1595 IDs. IJBC is updated from IJBB and includes 21294 images of 3531 objects. MegaFace aims at evaluating the face recognition performance at the million scale of distractors, which includes a large gallery set and a probe set. In this work, we use the Facescrub as the probe set of MegaFace as gallery.

4.2. Performance Comparisons between FC and F^{2}C

We choose 3 different backbones and evaluate the performance on 9 academic benchmarks between FC and F^{2}C using MS1MV2, Glint360k and Webface42M as training datasets. As shown in Table 1, F^{2}C can achieve comparable performance compared to FC. We also provide the average performance among these datasets and demonstrate it in the last column where F^{2}C is only lower than FC within 1%. Note that, the size of DCP is only 10% of the total face identities.

4.3. Ablation Studies

We conduct ablation studies of hyper parameters and settings of F^{2}C. Here we demonstrate the experiments on MS1MV2 using MobileFaceNet and ResNet50.

Single Loader or Dual Loaders? As mentioned in methodology section, dual loaders can improve the update efficiency of DCP. To evaluate the influence of loaders in F^{2}C, we use different combinations of identity-based and instance-based loaders and show the results in Table 2. The Small Datasets represent LFW, SLFW, CFP, CALFW, CPLFW, AGEDB and YTF in this subsection. We show the average accuracy on Small Datasets. Unless specified, TPR@FAR=1e-4 metric is used for IJBC and Megafce is FPR@FAR=1e-6 by default. Training with instance-based loader or identity-based loader can obtain comparable re-

Table 1: Evaluation results (%) on 9 face recognition benchmarks. All models are trained from scratch on MS1MV2, Glint360k and Webface42M. The TPR@FAR=1e-4 metric is used for IJBC. MegaFace is TPR@FAR=1e-6

Method	LFW	SLFW	CFP	CALFW	CPLFW	AGEDB	YTF	IJBC	MegaFace	Avg.
Training on MS1MV2										
FC-Mobile	99.04	98.80	96.94	94.37	88.37	96.73	97.04	92.29	90.69	94.92
F^{2}C-Mobile	98.93	98.57	97.16	94.53	87.80	96.47	97.24	91.06	89.30	**94.56**
FC-R50	99.78	99.55	98.80	95.76	92.01	98.13	98.03	95.74	97.82	97.29
F^{2}C-R50	99.50	99.45	98.46	95.58	90.58	97.83	98.16	94.91	96.74	**96.80**
Training on Glint360k										
FC-R50	99.83	99.71	99.07	95.71	93.48	98.25	97.92	96.48	98.64	97.67
F^{2}C-R50	99.71	99.53	98.30	95.23	91.60	97.88	97.76	94.75	96.73	**96.83**
Training on Webface42M										
FC-R100	99.83	99.81	99.38	96.11	94.90	98.58	98.51	97.68	98.57	98.15
F^{2}C-R100	99.83	98.80	99.33	95.92	94.85	98.33	98.23	97.31	98.53	**97.90**

Table 2: Evaluation of single or dual data loaders.ID.L, Ins.L and Dua.L represent id loader, instance loader and dual loaders respectively.

Backbone	Method	Small Datasets	IJBC	MegaFace
Mobile	ID.L	94.20	82.30	79.19
	Ins.L	94.24	89.30	86.40
	Dua.L	**95.29**	**91.06**	**89.30**
ResNet50	ID.L	96.70	91.75	93.65
	Ins.L	96.08	92.06	92.74
	Dua.L	**97.07**	**94.91**	**96.74**

Table 3: Evaluation of single net or dual nets.

Backbone	Method	Small Datasets	IJBC	MegaFace
Mobile	Single	93.90	88.07	82.69
	Dual	**95.29**	**91.06**	**89.30**
ResNet50	Single	95.55	92.26	92.98
	Dual	**97.07**	**94.91**	**96.74**

Table 4: Evaluation of the number of K.

Backbone	K	Small Datasets	IJBC	MegaFace
Mobile	1	95.19	90.75	88.31
	2	**95.29**	**91.06**	**89.30**
ResNet50	1	96.58	94.38	96.49
	2	**97.07**	**94.91**	**96.74**

Table 5: Evaluation the ratios within dual data loader. ResNet50 is used here.

Ins.L	ID.L	Small Datasets	IJBC	MegaFace
0	1	96.77	91.75	93.65
1	0	96.23	92.06	92.74
1	1	**97.08**	**94.91**	**96.74**
2	1	96.29	94.21	96.43
1	2	95.40	90.80	90.56

sults on small datasets. Instance-based loader outperforms identity-based loader on IJBC and MegaFace by a large margin. It could be explained that only using identity loader can not ensure all the images are sampled. Using dual data loaders can improve the performance compared with each single loader obviously, which is consistent to our analysis. Note that, to make fair comparison, the results are obtained with the same number of samples fed to the model, not with the same number of epoch.

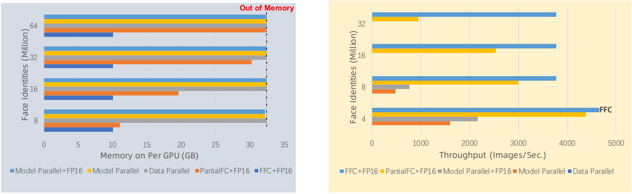
Single Net or Dual Nets? MoCo treates the two augmented images of the same image as positive samples and achieved impressive performance in unsupervised learning. Therefore, pictures with the same ID can naturally be regarded as positive samples, thus it is intuitive to use twin backbones in the same way as MoCo to generate the identities' centers and extract the face features respectively. However we intend to reduce the training cost further, so we compare the performance of single net to dual nets in Table 3. The dual nets performs better than single net on all datasets, which illustrates only using single net may fall into the trivial solution as explained in Semi-Siamese Training[9].

Exploring the Influence of K in DCP. K represents the number of the features that belong to the same identity. We evaluate the $K = 1$ and $K = 2$ in Table 4. As the features in DCP represent the category centers, an intuitive sense is that a larger K can provide more reliable center estimation. The experiments results also support our intuition. However we must make a trade-off between performance and storage. A larger K means better performance at the cost of GPU memory and communication among severs. Therefore, we set $K = 2$ in DCP by default.

Ratios within dual data loader. We set the ratio of the size between instance-based and identity-based loaders as 1:1 by default. To further explore the influence of the ratios within dual data loader, we show the experiments in Ta-

Table 6: Comparisons to state-of-the-art methods. To make fair comparison, Partial-FC, VFC, DCQ and **F^{2}C** only use 1% of identities of MS1M for training. Megaface refers to rank-1 identification. IJBC is TPR@FAR=1e-4. The lower-boundary results are excerpted from VFC paper. The upper-boundary results are reproduced by us.

Method	CALFW	CPLFW	SLFW	YTF	CFP	IJBC	MegaFace
lower-boundary	87.43	75.45	93.52	93.78	91.66	65.19	79.28
upper-boundary	95.75	90.85	99.55	97.76	98.39	95.48	97.56
N-pair[31]	87.32	72.80	92.28	92.62	-	61.75	82.56
Multi-similarity[41]	85.40	73.60	91.03	92.76	-	57.82	76.88
TCP[22]	88.05	76.00	93.23	93.92	93.27	43.58	88.18
Partial-FC[1]	95.40	90.33	99.28	97.76	98.13	94.40	94.13
VFC[20]	91.93	79.00	96.23	95.08	95.77	70.12	93.18
DCQ[19]	95.38	88.92	99.23	97.71	98.16	92.96	95.21
F^{2}C	95.25	89.38	99.23	97.76	98.25	92.31	94.25



(a) Comparison of GPU memory Occupancy (GB).

(b) Comparisons of Throughput (Images/Sec.).

Figure 3: Visualizations of the hardware resource occupancy of different training methods.

ble 5. We utilize ResNet50 as backbone to train MS1MV2 dataset. We find that the default ratio within dual data loader achieves the highest results on most datasets, especially on challenging IJBC and MegaFace.

4.4. Comparisons with SOTA Methods.

We compare our F^{2}C to other 6 state-of-the-art methods and show the results in Table 6. We can observe that F^{2}C outperforms lower-boundary, N-pair, Multi-similarity, and TCP by a large margin, especially on IJBC and MegaFace datasets. As claimed in VFC [20], Upper-boundary represents training with normal FC using the 100% face identities. F^{2}C has a little degradation of performance than upper-boundary. It can also achieve comparable results with Partial-FC, but Partial-FC requires hardware space to store the total identities' centers while VFC doesn't. However the performance of VFC drops obviously compared to F^{2}C.

Visualizations of Resource Cost and Training Efficiency. The GPU memory occupancy and throughput are two crucial factors to evaluate the practicability of a method in distributed parallel training. To better understand the efficiency of F^{2}C, Figure 3 visualizes the GPU memory occupancy and throughput of F^{2}C and other training methods. The results are obtained on a 8 V100 32G GPUs. GPU memory occupancy is illustrated in Figure 3a, Data Parallel and Model Parallel are out-of-memory (OOM) when the identities reach to 16 million. The memory of Partial-FC increases with growth of the identities and it also OOM when the identities reach to 32 million. Besides, we show the throughput comparisons in Figure 3b, only F^{2}C can keep the high-level throughput among different number of identities. Therefore, the proposed F^{2}C is practical in ultra-large-scale face recognition task.

5. Conclusion

In this paper, we propose an efficient training approach F^{2}C for ultra-large-scale face recognition training, the main innovation is Dynamic Class Pool (DCP) for store and update of face identities' feature as an substitute of FC and dual loaders for helping DCP update efficiently. The results of comprehensive experiments and analysis show that our approach can reduce hardware cost and time for training as well as obtaining comparable performance to state-of-the-art FC-based methods.

Broader impacts. The proposed method is validated on face training datasets due to the wide variety, the scheme could be expanded to other datasets and situations. However, it does not contain any studies involving affecting ethics or human rights performed by any of the authors.

6. Acknowledge

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-008). This work is supported by Alibaba Group through Alibaba Research Intern Program, the Chinese National Natural Science Foundation Projects 62106264 and 62176165. We thank Google TFRC for supporting us to get access to the Cloud TPUs, CSCS (Swiss National Supercomputing Centre) for supporting us to get access to the Piz Daint supercomputer, TACC (Texas Advanced Computing Center) for supporting us to get access to the Longhorn supercomputer and the Frontera supercomputer, and LuxProvide (Luxembourg national supercomputer HPC organization) for supporting us to get access to the MeluXina supercomputer.

References

[1] Xiang An, Xuhan Zhu, Yang Xiao, Lan Wu, Ming Zhang, Yuan Gao, Bin Qin, Debing Zhang, and Ying Fu. Partial fc: Training 10 million identities on a single machine. *arXiv preprint arXiv:2010.05222*, 2020.

[2] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE international joint conference on biometrics (IJCB)*, pages 464–473. IEEE, 2017.

[3] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 67–74, 2018.

[4] Shiming Chen, Ziming Hong, Yang Liu, Guo-Sen Xie, Baigui Sun, Hao Li, Qinmu Peng, Ke Lu, and Xinge You. Transzero: Attribute-guided transformer for zero-shot learning. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, 2022.

[5] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018.

[6] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 122–131, 2021.

[7] Shiming Chen, Guo-Sen Xie, Qinmu Peng, Yang Liu, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[9] Hang Du, Hailin Shi, Yuchi Liu, Jun Wang, Zhen Lei, Dan Zeng, and Tao Mei. Semi-siamese training for shallow face learning. In *European Conference on Computer Vision*, pages 36–53. Springer, 2020.

[10] Joshua Goodman. Classes for fast maximum entropy training. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 1, pages 561–564. IEEE, 2001.

[11] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Guosheng Hu, Xiaojiang Peng, Yongxin Yang, Timothy M Hospedales, and Jakob Verbeek. Frankenstein: Learning deep face representations using small data. *IEEE Transactions on Image Processing*, 27(1):293–303, 2017.

[15] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[16] Bong-Nam Kang, Yonghyun Kim, and Daijin Kim. Pairwise relational networks for face recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 628–645, 2018.

[17] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[18] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[19] Bi Li, Teng Xi, Gang Zhang, Haocheng Feng, Junyu Han, Jingtuo Liu, Errui Ding, and Wenyu Liu. Dynamic class queue for large scale face recognition in the wild, 2021.

[20] Pengyu Li and Lei Zhang BiaoWang. Virtual fully-connected layer: Training a large-scale face recognition dataset with limited computational resources.

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[22] Yu Liu, Guanglu Song, Jing Shao, Xiao Jin, and Xiaogang Wang. Transductive centroid projection for semi-supervised large-scale recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–86, 2018.

[23] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018.

[24] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017.

[25] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[26] Xiaojiang Peng, Kai Wang, Zhaoyang Zeng, Qing Li, Jianfei Yang, and Yu Qiao. Suppressing mislabeled data via grouping and self-attention. In *European Conference on Computer*

Vision, pages 786–802. Springer, 2020.

[27] Xiangyu Peng, Kai Wang, Zheng Zhu, and Yang You. Crafting better contrastive views for siamese representation learning. *arXiv preprint arXiv:2202.03278*, 2022.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[29] S. Sengupta, J. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[32] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[34] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 765–780, 2018.

[35] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[36] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[37] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. Suppressing uncertainties for large-scale facial expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6906, 2020.

[38] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020.

[39] Kai Wang, Shuo Wang, Jianfei Yang, Xiaobo Wang, Baigui Sun, Hao Li, and Yang You. Mask aware network for masked face recognition in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1456–1461, 2021.

[40] Kai Wang, Bo Zhao, Xiangyu Peng, Yang You, et al. Cafe: Learning to condense dataset by aligning features. *CVPR2022*, 2022.

[41] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[42] Xiaobo Wang, Shuo Wang, Cheng Chi, Shifeng Zhang, and Tao Mei. Loss function search for face recognition. In *International Conference on Machine Learning*, pages 10029–10038. PMLR, 2020.

[43] Xiaobo Wang, Shuo Wang, Jun Wang, Hailin Shi, and Tao Mei. Co-mining: Deep face recognition with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9358–9367, 2019.

[44] Xiaobo Wang, Shuo Wang, Shifeng Zhang, Tianyu Fu, Hailin Shi, and Tao Mei. Support vector guided softmax loss for face recognition. *arXiv preprint arXiv:1812.11317*, 2018.

[45] Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z Li. Ensemble soft-margin softmax loss for image classification. *arXiv preprint arXiv:1805.03922*, 2018.

[46] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011.

[47] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[48] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Z Li. Large scale similarity learning using similar pairs for person verification. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[49] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[50] Xingcheng Zhang, Lei Yang, Junjie Yan, and Dahua Lin. Accelerated training for massive classification via dynamic class selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[51] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5, 2018.

[52] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.

[53] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. *arXiv preprint arXiv:2103.04098*, 2021.